

5. Lovtsov D.A Legal informatics: studies. manual / D.A. Lovtsov. — M.: Ross. akad. pravosudyia, 2007. [in Russian]
6. Morozov A.V. Conception of legal informatization : studies. manual / A.V. Morozov. — M.: VHUu (RPA Myniusta Russia), 2015. [in Russian]
7. Morozov A.V. Legal questions of access to information: studies. manual / A.V. Morozov, L.V. Fylatova. — M.: VHUI (RPA of Myniusta of Russia), 2015. [in Russian]
8. Legal informatics: studies. Manual: [release M.M. Rassolova]. — M.: «Manuskrypt», 1992. [in Russian]
9. Chubukova S.H. Bases of legal informatics (legal and mathematical questions of informatics): studies. manual / S.H. Chubukova, V.D. Elkyn, [release M.M. Rassolova]. — M.: «Kontrakt», 2004. [in Russian]
10. Venherov A.B. Synergetics, legal science, right/ A.B. Venherov // Sovetskoe hosudarstvo y pravo (Soviet state and right). — 1986. — № 10. — S. 364. [in Russian]
11. Artiukhov V.V. The general theory of the systems: organization, stability, variety, crises / V.V. Artiukhov. — M.: Knyzhnyi dom «LYBRAKOM», 2009. — 224 s. [in Russian]
12. Komarov S.A. General description of mechanism of the legal adjusting // is the General theory of the state and right : the Academic course in 2 volumes [editor. M.N. Marchenko]. T. 2: Teoryia prava. M.: Yzdatelstvo «Zertsalo», 2000. S. 439. [in Russian]

Статтю подано до редакції 21.01.2019 р.

УДК 331.103, 65.01

DOI: 10.33111/mise.97.18

Потапенко С. Д., к.е.н.,
доцент кафедри інформаційного менеджменту,
Київський національний економічний університет
імені Вадима Гетьмана

Potapenko S. D., Candidate of Economic Sciences,
Associate Professor of the Information Management Department,
Kyiv National Economic University named after Vadym Hetman

ОСОБЛИВОСТІ ОРГАНІЗАЦІЇ КОЛЕКТИВНОГО СЕРВЕРНОГО ПРОГРАМУВАННЯ У СИСТЕМІ МЕНЕДЖМЕНТУ СТВОРЕННЯ ІНТЕРНЕТ-РЕСУРСІВ

FEATURES OF THE ORGANIZATION OF COLLECTIVE SERVER PROGRAMMING IN THE MANAGEMENT SYSTEM FOR CREATING INTERNET RESOURCES

Анотація. Управління процесом створення програмного забезпечення є одним з актуальних напрямів діяльності менеджерів, які працюють в інформаційній галузі. У статті розглянуто деякі аспекти організації

роботи колективів розробників з урахуванням специфіки серверного програмування Інтернет-ресурсів. У цей процес постійно залучаються нові фахівці та формуються нові передумови організації колективної праці у проектах, що пов'язані із залученням мережі Інтернет у здійснення комерційної діяльності різноманітних суб'єктів господарювання. Зростання складності задач, які вирішуються, підвищення вимог до компетенцій фахівців, розвиток можливостей інструментарію створення Інтернет-ресурсів вимагають від менеджерів різноманітних проектів, що орієнтуються на використання мережі Інтернет, замислюватись над необхідністю якісної організації праці відповідних фахівців. Виконано аналіз останніх досліджень і публікацій, що стосуються розвитку теорії менеджменту, зокрема у питанні визначення функцій менеджменту їх реалізації. Оскільки, з точки зору сучасної науки управління, однією з функцій менеджменту є організація виконання робіт — то ефективний менеджер повинен опікуватись питанням забезпечення даної функції у системі управління відповідних виробничих процесів. У статті розглянуто два підходи до організації робіт, що пов'язані з серверним програмуванням Інтернет-ресурсів — без поділу та з поділом тексту програми на незалежні одиниці трансляції мови програмування. Проаналізовано можливості кожного з підходів. Розкрито переваги застосування підходу до серверного програмування, що базується на розподілі тексту програму на окремі одиниці трансляції, та визначено переваги, які можуть отримати колективи розробників. Визначено перспективи подальшого розвитку запропонованої ідеї. Продемонстровано приклад реалізації задачі та зроблено відповідні висновки з приводу організації виконання відповідних робіт.

Ключові слова: серверне-програмування; Інтернет-ресурс; мова програмування C++; одиниця трансляції; глобальна змінна; зв'язування ідентифікаторів.

Abstract. Managing the software creation process is one of the most important activities of managers working in the information industry. The article deals with some aspects of organizing the work of development teams, taking into account the specifics of server programming of Internet resources. This process constantly involves new specialists and creates new prerequisites for organizing collective work in projects related to the involvement of the Internet in the commercial activities of various business entities. The growing complexity of problems solved, the increase of requirements to the skill, the capability of the tools to create online resources requires managers of various projects oriented at the use of the Internet to reflect on the need for high-quality labour organization relevant experts. The analysis of the latest research and publications related to the development of management theory, in particular in determining the functions of management of their implementation. Since, from the point of view of modern management science, one of the functions of management is the organization of work performance, an effective Manager should deal with the issue of ensuring this function in the management system of the corresponding production processes. The article considers two approaches to organizing work related to server programming of Internet resources-without separation and with the division of the program text into independent translation units of the programming language. The possibilities of each approach are analyzed. The advantages of applying an approach to server programming based on the distribution of the program text into separate translation units are revealed, and the advantages that can be obtained by development teams are determined. The prospects for further development of the proposed idea are determined. An example of the implementation of the task is demonstrated and appropriate conclusions are made about the organization of the relevant work.

Key words: server side programming; Internet resource; C++ programming language; translation unit; global variable; identifier binding.

Вступ. Стрімкий розвиток інформаційного простору сучасного світу спричиняє появу нових інформаційних ресурсів і стимулює розвиток тих, що вже існують. Зокрема, у цей процес постійно залучаються нові фахівці та формуються нові передумови організації колективної праці у проєктах, що пов'язані із залученням мережі Інтернет у здійснення комерційної діяльності різноманітних суб'єктів господарювання. Зростання складності задач, які вирішуються, підвищення вимог до компетенцій фахівців, розвиток можливостей інструментарію створення Інтернет-ресурсів вимагають від менеджерів різноманітних проєктів, що орієнтуються на використання мережі Інтернет, замислюватись над необхідністю якісної організації праці відповідних фахівців. Оскільки подібна задача безпосередньо стосується питань якісного менеджменту створення різноманітних Інтернет-ресурсів — то особливості подібних процесів обрані для розгляду у даній статті.

Формулювання мети статті. Менеджмент будь-яких виробничих процесів визначається вмістом функцій системи управління ними. Як буде показано нижче, різні дослідники дають різне тлумачення складу функцій менеджменту, але, у той же час, є спільна риса, яку не можна не відмітити — наявність функції, яка у тому або іншому вигляді, завжди присутня майже у всіх дослідників. Такою функцією менеджменту — є функція організації. Дійсно, однією із задач керівника є формування передумов здійснення праці, а саме вжиття всіх необхідних заходів, які потребуються для здійснення виробничих завдань тощо. Саме менеджеру необхідно приймати рішення про вибір потрібних технологій, залучення необхідних ресурсів та обрання інструментів, використання яких забезпечить якісну реалізацію відповідних проєктів. Процеси створення Інтернет-ресурсів мають низку притаманних їм особливостей. Через це, висвітлення даних особливостей у контексті функції організації менеджменту обрано у якості мети даної статті.

Аналіз останніх досліджень і публікацій. Огляд останніх досліджень у сфері менеджменту дозволив визначити, що сучасний стан академічного розвитку теорії менеджменту у значній мірі обумовлений фундаментальними дослідженнями, які проводились на початку 20-го сторіччя А. Файолем, Г. Емерсоном, Ф. Тейлором, Г. Фордом та іншими. У наш час менеджмент, як наука, має власний науковий інструментарій, виокремлено актуальні напрями її розвитку — стратегічний менеджмент, інформаційний менеджмент, інвестиційний менеджмент, фінансовий менеджмент, інноваційний менеджмент, антикризовий менедж-

мент, виробничий менеджмент тощо, проводяться наукові дослідження різноманітних аспектів теорії управління. У той же час, не зважаючи на активний розвиток прикладних застосувань менеджменту, не згасає зацікавленість до розвитку його фундаментальних основ. У науковій і навчальній літературі різними авторами постійно виконується уточнення формулювання задач менеджменту, його мети, визначення функцій та особливостей їх реалізації в сучасних умовах.

Розглянемо низку популярних визначень функцій менеджменту. Зокрема, у [5] зазначено, що процес управління — це сукупність безперервних, послідовних, взаємопов'язаних дій по формуванню і використанню ресурсів організації для досягнення нею своїх цілей. Такі дії важливі для успіху організації і є функціями менеджменту. Таким чином, процес управління є загальною сумою всіх його функцій, кожна з яких є важливою і може розглядатись, за відповідних передумов, окремо від інших.

У період зародження теорії менеджменту до таких функцій відносили: передбачення; організацію; розпорядництво; координування та контроль [6]. Такі функції є фундаментальними для багатьох підприємств і в певній мірі притаманні кожній організації.

На думку інших авторів сучасна концепція складу функцій менеджменту повинна бути дещо іншою і включати в себе: планування; організацію; керівництво; координацію та контроль [3]. Або, більш розгорнута класифікація, яку також можна зустріти у того ж автора: планування; організація; регулювання; координація; мотивація; керівництво; контроль. З такою думкою не можна не погодитись, так як сучасні підприємства значно відрізняються за розмірами, рівнем технологій, сферою діяльності. Отже обирати кількість потрібних для організації функцій менеджменту дійсно потрібно індивідуально у кожному окремому випадку.

Дещо у зміненому стані пропонується визначати функції менеджменту у [4]. До функцій менеджменту автор, у своїй роботі, пропонує відносити: прогнозування; планування; керівництво; координацію та контроль.

Актуальним до вимог сьогодення можна вважати склад функцій менеджменту, який запропонований у [5]. Зокрема, таким функціями слід вважати: планування; організацію; мотивацію; контроль; координацію та регулювання. Такий набір функцій найповніше відображає склад задач, які постійно потрібно вирішувати фахівцям підприємств високотехнологічних галузей, підприємств сфери інформаційних технологій, підприємств, що працюють у високо-конкурентному середовищі тощо.

Найрозгорнутіший перелік функцій менеджменту, серед робіт які розглядаються, можна зустріти у [1]. До таких функцій автори відносять: формування цілі; формування стратегії; планування; регулювання; мотивацію; стимулювання; гуманізацію; корпоративність; організацію; координацію; контроль та оцінку.

Нарешті найпопулярнішою, серед багатьох науковців, є класифікація функцій менеджменту, яка запропонована у [2]. До таких функцій автори відносять: планування; організацію; мотивацію та контроль. Такий перелік функцій менеджменту є актуальним для більшості сучасних підприємств.

Можна відмітити, що функція організації виділяється майже всіма авторами, що, у певному сенсі, доводить її важливість.

Викладення основного матеріалу. Оскільки матеріал статі спрямований на висвітлення особливостей розроблення Інтернет-ресурсів у контексті функції організації, яка є частиною системи менеджменту даного процесу, — то розглянемо деякі особливості роботи колективів розробників, що залучені у даний процес. Враховуючи певну специфіку діяльності даних фахівців можна надати змістовний опис деяким діям з визначенням їх переваг, недоліків і способів усунення останніх. Сучасне розроблення Інтернет-ресурсів тяжіє до реалізації такої моделі архітектури програмного забезпечення, як «тонкий клієнт». Дана модель вимагає розподілу виконання обчислювального навантаження на серверну та клієнтську частини. При цьому серверна частина — виконує доступу до даних, забезпечує здійснення необхідних обчислень. У той же час клієнтська частина необхідна для формування запитів до Інтернет-ресурсу та відображення отриманого результату виконання даних запитів.

Потрібно зазначити, що деяке обчислювальне навантаження також може бути зосереджене в межах клієнтської частини Інтернет-ресурсу. Такий випадок не розглядається у даній статі. Матеріал статі орієнтовано на розгляд традиційної форми функціонування Інтернет ресурсів. Під традиційною формою функціонування мається на увазі обслуговування запитів до ресурсу методами GET і POST, що використовуються для надсилання параметрів запитів та відправлення сформованих відповідей у формі програмного коду розмітки гіпертекстових електронних документів мовою HTML без залучення додаткових можливостей.

Розглянемо деякі особливості, що пов'язані з отриманням запитів і формуванням відповідей на них у межах серверної складової деякого гіпотетичного Інтернет-ресурсу. Над створення

серверної складової одночасно можуть працювати кілька розробників. Це може бути обумовлено специфікою оброблення передачі даних різними методами з розподілом такої задачі між фахівцями, які спеціалізуються на використанні відповідних методів. Отже, специфіка засобів, які застосовуються у процесі створення Інтернет-ресурсу, визначає спеціалізацію фахівця. Але, у той же час, можливі ситуації колективної роботи з одними і тими ж даними, файлами початкових кодів тощо. Також, така особливість пов'язана зі специфікою використання обраних керівником мов програмування — наприклад, концепції реалізації точки входження у програму. Через те, що така задача є природною, їй часто не приділяється достатньо уваги і виникають різноманітні колізії у виробничому процесі, які, здебільшого, пов'язані з різноманітними конфліктними ситуаціями, що усуваються адміністративним шляхом.

Розглянемо подібний випадок на основі простої задачі, яка, тим не менш, містить у собі весь типовий спектр завдань, вирішення яких потрібно реалізувати у межах будь-якого Інтернет-ресурсу із залученням як серверної, так і клієнтської складової. Нехай такою задачею є складання двох чисел a та b . Для її вирішення, як і для будь-якої іншої подібної задачі, буде потрібним формування запиту до Інтернет-ресурсу з боку браузера, надсилення запиту браузером до серверу, прийняття запиту сервером, аналізу вмісту запиту, формування відповіді на запит, відправлення відповіді сервером до браузера, прийняття відповіді браузером і відображення ним результату.

З погляду серверного програмування такі завдання, як прийняття запиту, аналізу його вмісту, формування відповіді та її відправлення — часто розглядаються, як єдине ціле. Іноді це викликано потребою передати задачу програмування виконання даних завдання окремому розробнику, у інших випадках — це вимога відповідних засобів, наприклад, мов програмування.

Нехай мовою програмування серверної складової Інтернет-ресурсів є мова C++. Однією з особливостей даної мови програмування є можливість розподілу початкового коду, який стосується спільних задач, на кілька одиниць трансляції — файлів, у яких міститься даний програмний код, що є природним для даного засобу. Але, разом з тим, даний інструмент може бути застосований і в умовах відсутності такого розподілу.

Наприклад, в умовах зазначеної задачі, без розподілу на окремі одиниці трансляції — початковий код її вирішення може мати вигляд програмного коду 1.

Програмний код 1. Вирішення задачі без розподілу на одиниці трансляції

```
1: #include <ps/app.h>
2: #include <ps/cgi.h>
3: #include <ps/signal.h>
4:
5: // Об'єкт, що забезпечує обслуговування виконання запитів
6: static std::unique_ptr<ps::cgi::Server> server;
7:
8: // Оброблювач сигналів
9: static void handler(int)
10: {
11:     if (server) server->stop();
12: }
13:
14: // Вхідження у програму
15: int main()
16: {
17:     return ps::app([] ()
18:     {
19:         // Правила оброблення запитів
20:         ps::cgi::Rules rules;
21:
22:         // Формування вмісту початкової сторінки
23:         rules.defGet
24:         (
25:             [] (const ps::cgi::Request &request)
26:             {
27:                 if (request.query().size() == 0)
28:                 {
29:                     auto &envs = request.envs();
30:                     auto uri = envs.find("DOCUMENT_URI");
31:                     return uri != envs.end() && uri->second == "/";
32:                 }
33:                 return false;
34:             },
35:             [] (ps::cgi::Request &request)
36:             {
37:                 request.cout()
38:                 << "<form action='/?task=addition'>"
39:                 "<input type=hidden name=task value='addition'>"
40:                 "a=<input type=text name=a>"
41:                 "b=<input type=text name=b>"
42:                 "<input type=submit>"
43:                 "</form>";
44:             },
45:             "content", "site.tpl"
46:         );
47:
48:         // Виконання обчислення
49:         rules.defGet
```

```

50: (
51:     [] (const ps::cgi::Request &request)
52:     {
53:         return request.query().has("task", "addition");
54:     },
55:     [] (ps::cgi::Request &request)
56:     {
57:         auto a = ps::to_double(request.query().string("a"));
58:         auto b = ps::to_double(request.query().string("b"));
59:         request.cout()
60:             << "a=" << a << "<br>"
61:             << "b=" << b << "<br>"
62:             << "a+b=" << a + b << "<br>"
63:             << "<a href='/'>Головна сторінка</a>";
64:     },
65:     "content", "site.tpl"
66: );
67:
68: // Визначення правил оброблення сигналів
69: SIGNAL_IGNORE(SIGPIPE);
70: SIGNAL(SIGTERM, handler);
71: SIGNAL(SIGINT, handler);
72: SIGNAL(SIGHUP, handler);
73:
74: // Запуск сервера виконання запитів
75: server = std::make_unique<ps::cgi::Server>("127.0.0.1:3456");
76: server->start(rules);
77:
78: return EXIT_SUCCESS;
79: });
80: }

```

Даний приклад демонструє вирішення зазначеної задачі через виконання двох запитів — запиту на створення форми введення значень *a* та *b* і запиту на виконання складання даних чисел. Приклад штучний і потрібен лише для демонстрації зазначених аспектів серверного програмування. Програмний код оформлений у вигляді одного файлу, що ускладнює сумісне редагування одних і тих самих ділянок кількома розробниками. Натомість інший підхід позбавлений такого недоліку. Використання таких особливостей мови програмування C++, як застосування глобальних змінних зі зв'язуванням ідентифікаторів, яке виходить за межі одиниць трансляції, а також зі зв'язуванням ідентифікаторів, яке не виходить за межі одиниць трансляції при умові, що дані змінні є екземплярами класів, у конструкторах яких визначаються правила оброблення запитів до Інтернет-ресурсу — відкривають можливості до більш гнучкої організації серверного програмування подібних задач. Останнє твердження підкріплю-

ється прикладом програмного коду, у якому визначення правил оброблення запитів здійснено через оголошення глобальних змінних, зв'язування ідентифікаторів яких не виходить за межі одиниць трансляції. Визначення таких змінних здійснюється через застосування спеціальних макросів-функцій фрагмент початкового коду яких подано у програмному коді 2. Дані макроси-функції використовують деякі службові макроси-об'єкти, які надає компілятор мови програмування [7]. Використання службових макросів-функцій дозволяє визначити поточну позицію у початковому коді і через це надає змогу отримати приховане службове ім'я відповідного глобального об'єкту.

Програмний код 2. Допоміжні макроси-функції визначення правил оброблення запитів

```

1: #define BASIC_NAMECOMB(a,b) a##b
2: #define NAMECOMB(a,b) BASIC_NAMECOMB(a,b)
3: #define WEB_RULE(...) static ::ps::cms::DefWebRule \
4:   NAMECOMB(ps cms_web_rule , _LINE_)(__VA_ARGS__);
5: #define RULE_GET(...) WEB_RULE \
6:   ([ (ps::cgi::Rules &rules) { rules.defGet(__VA_ARGS__); } )
7: #define CONDITION [ (const ps::cgi::Request &request)
8: #define ANSWER [ (ps::cgi::Request &request)

```

Оголошення глобальних змінних, зв'язування ідентифікаторів яких не виходить за межі одиниць трансляції, базується на створенні екземплярів об'єктів спеціально визначеного класу [8]. Даний клас містить лише конструктор. Зразок скороченого прототипу названого класу подано у початковому коді 3.

Програмний код 3. Скорочений прототип класу об'єкту визначення правил оброблення запитів

```

1: namespace ps
2: {
3:   namespace cms
4:   {
5:     class DefWebRule
6:     {
7:     public:
8:       using Action = std::function<void(ps::cgi::Rules &rules)>;
9:       DefWebRule(const Action &action);
10:    };
11:  }
12: }
13:

```

Конструктор даного класу має утилітарне значення і використовується лише з метою отримання можливості централізованого збереження правил оброблення запитів до Інтернет-ресурсів.

Спираючись на описану концепцію можна у доволі простий спосіб здійснити перехід від технології, яка не передбачає природнього поділу на одиниці трансляції мови програмування, до технології серверного програмування оброблення запитів до Інтернет-ресурсів з активним залученням саме природнього поділу на одиниці трансляції. Під природнім поділом мається на увазі відсутність потреби в активному залученні різноманітних функцій-регістраторів, в описі складових елементів через використання зовнішніх налаштувань, в особливому конфігуруванні збиральника програмного забезпечення в залежності від окремих вимог тощо. Натомість отримуємо можливість здійснювати серверне програмування через просте збагачення бази правил оброблення запитів. Особливості визначення правил оброблення, особливості їх збереження та вибору для застосування — є окремим матеріалом для висвітлення і не розглядається у даній статі.

Розглянемо поділ програмного коду задачі складання двох чисел на окремі одиниці трансляції. У зазначених умовах точка входження у програму буде мати вигляд, що продемонстрований к програмному коді 4.

Програмний код 4. Точка входження у програму

```
1: #include <ps/app.h>
2: #include <ps/cms.h>
3:
4: // Вхідження у програму
5: int main()
6: {
7:     return ps::app(I 0
8:     {
9:         ps::cms::start("127.0.0.1:3456");
10:        return EXIT_SUCCESS;
11:    });
12: }
```

Оскільки складовими елементами вирішення задачі є окремі одиниці трансляції, то запити на створення форми введення значень a та b і на виконання складання даних чисел також мають вигляд окремих файлів. Зокрема початковий код на створення форми введення чисел подано у програмному коді 5, а початковий код на виконання складання чисел — у програмному коді 6.

Програмний код 5. Формування форми введення аргументів обчислення

```
1: #include <ps/cms.h>
2:
3: // Формування вмісту початкової сторінки
```

```

4: RULE_GET
5: (
6:   CONDITION
7:   {
8:     if (request.query().size() == 0)
9:     {
10:      auto &envs = request.envs();
11:      auto uri = envs.find("DOCUMENT_URI");
12:      return uri != envs.end() && uri->second == "/";
13:    }
14:    return false;
15:  },
16:  ANSWER
17:  {
18:    request.cout()
19:    << "<form action=/'?task=addition'>"
20:    "<input type=hidden name=task value='addition'>"
21:    "a=<input type=text name=a>"
22:    "b=<input type=text name=b>"
23:    "<input type=submit>"
24:    "</form>";
25:  },
26:  "content", "site.tpl"
27: )

```

Програмний код 6. Виконання обчислення

```

1: #include <ps/cms.h>
2:
3: // Виконання обчислення
4: RULE_GET
5: (
6:   CONDITION
7:   {
8:     return request.query().has("task", "addition");
9:   },
10:  ANSWER
11:  {
12:    auto a = ps::to_double(request.query().string("a"));
13:    auto b = ps::to_double(request.query().string("b"));
14:    request.cout()
15:    << "a=" << a << "<br>"
16:    << "b=" << b << "<br>"
17:    << "a+b=" << a + b << "<br>"
18:    << "<a href=/'>Головна сторінка</a>";
19:  },
20:  "content", "site.tpl"
21: )

```

Програмний код 1, 4, 5 та 6 приведені у повному обсязі з метою демонстрації наочності зазначеної ідеї. Підхід до організації процесу серверного програмування Інтернет-ресурсів на основі поділу програмного коду задачі на окремі незалежні одиниці

трансляції дає змогу нівелювати загрозу появи конфліктних ситуацій, що пов'язані зі спільним редагуванням початкових кодів кількома розробниками одночасно. Кожний розробник отримує свій фрагмент вирішення задачі і зосереджується лише на ньому. У свою чергу, менеджер проекту отримує якісний інструмент організації процесу серверного програмування, що є невід'ємною частиною системи менеджменту створення Інтернет-ресурсу в цілому. Розглянута технологія поділу програмного коду задачі на окремі одиниці трансляції також дозволяє здійснити природний перехід до формалізації таких функцій менеджменту управління створення програмного продукту, як планування та контроль виконання робіт, що заслуговує на окремий розгляд.

Висновки. На прикладі розглянутих аспектів бекенд-програмування, можна стверджувати, що для якісного управління процесом створення web-сайтів є необхідним володіння сучасним інструментарієм програмування. Володіння даними прикладними інструментами дозволяє якісно організувати виконання робіт, що є невід'ємною частиною формування системи управління відповідних процесів. Зокрема, через зазначені міркування, володіння подібним інструментарієм повинно бути складовим елементом наряду підготовки майбутніх фахівців з менеджменту за відповідним профілем тощо.

Література

1. Менеджмент (Современный российский менеджмент). Учебник / Авт. кол. Ф. М. Русинов, М. Л. Разу, В. А. Денисов и др.; Под ред. Ф. М. Русинова, М. Л. Разу. М.: ФБК-Пресс, 1999. 504 с.
2. Мескон М., Альберт М., Хедоури Ф. Основы менеджмента. М.: Издательство «Дело», 1997. 704 с.
3. Михалева Е. П. Менеджмент : конспект лекций. — 2е изд., перераб. и доп.. М.: Юрайт, 2010. 191 с.
4. Орлов А. И. Менеджмент. Учебник. М.: Издательство «Измуд», 2003. 298 с.
5. Рульев В. А., Гуткевич С. О. Менеджмент. Навч. посіб.. К.: Центр учбової літератури, 2011. 312 с.
6. Файоль А., Эмерсон Г., Тейлор Ф., Форд Г. Управление это наука и искусство. М.: Республика, 1992. 351 с.
7. Kernigan B. W., Ritchie D. M. The C programming language. Englewood Cliffs: Prentice Hall, 1978. 228 p.
8. Stroustrup B. The C++ Programming Language Fourth Edition. Ann Arbor: Addison-Wesley, 2013. 1346 p.

References

1. Rusynov, F. M. & Razu, M. L. & Denysov, V. A. & etc (1999). Menedzhment (Sovremennij rossyjskij menedzhment). FBK-Press. M. [In Russian]
2. Meskon, M. & Albert, M. & Xedoury, F. (1997). Osnovi menedzhmenta. Yzdatel'stvo «Delo». M. [In Russian]
3. Myxaleva, E. P. (2010). Menedzhment: konspekt lekcyj. Yurajt. M. [In Russian]
4. Orlov, A. Y. (2003). Menedzhment. Yzdatel'stvo «Yzumrud». M. [In Russian]
5. Rulev, V. A. & Hutkevych, S. O (2011). Menedzhment. Centr uchbovoyi literatury. K. [In Ukrainian]
6. Fajol, A. & Jemerson, G. & Tejlор, F. & Ford, G. (1992). Upravlenie jeto nauka i iskustvo. Respublika. M. [In Russian]
7. Kernigan, B. W. & Ritchie, D. M. (1978). The C programming language. Prentice Hall. Englewood Cliffs. [In English]
8. Stroustrup, B. (2013). The C++ Programming Language (Fourth Edition). Addison-Wesley. Ann Arbor. [In English]

Статтю подано до редакції 17.01.2019 р.

УДК 004.67

DOI: 10.33111/mise.97.19

Ситник Н. В., к.е.н.,

доцент кафедри інформаційних систем в економіці
ДВНЗ «КНЕУ імені Вадима Гетьмана»

Кузик М. М.

магістр спеціалізації «Інформаційні управляючі системи та технології»
Інституту інформаційних технологій в економіці
ДВНЗ «КНЕУ імені Вадима Гетьмана»

Sytnyk N. V.

PhD in Economics, Associate Professor of the Economics
Information Systems Department,

Kuzyk M. M., Master Student «Information management system»
Kyiv National Economic University named after Vadym Hetman

**КОМП'ЮТЕРНА ПІДТРИМКА УПРАВЛІННЯ
ПРОЦЕСОМ ПОСЕЛЕННЯ В СТУДЕНТСЬКОМУ
МІСТЕЧКУ ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДІВ**

**COMPUTER SUPPORT OF THE SETTLEMENT
PROCESS MANAGEMENT IN THE CAMPUS
OF HIGHER EDUCATION INSTITUTIONS**